

Isofill File

```
# Import the modules needed for the tutorial
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

# Initial VCS.
v = vcs.init()

# Show the list of persistent isofill graphics methods.
v.show('isofill')

# Assign the variable "cf_asd" to the persistent 'ASD' isofill graphics methods.
cf_asd = v.getisofill( 'ASD' )

# Plot the data using the above boxfill graphics method.
v.plot( data, cf_asd )
print ""
print "Press the Return key to see next plot."
sys.stdin.readline()

# List the 'ASD' isofill graphics methods attributes.
cf_asd.list()

# change the isofill levels and the color indices.
cf_asd.levels = ( [0,20],[20,40],[50,60],[60,70],[70,80],[90,100])
cf_asd.fillareacolors=( [22,44,66,88,110,132])

# change the isofill levels and the color indices.
cf_asd.levels = ([0,20,25,30,35,40],[70,75,80], [90,100])
cf_asd.fillareacolors=( [22,44,66,88,110,132,152,174])
print "Press the Return key to see next plot."
sys.stdin.readline()

# Change the levels back to the default settings.
cf_asd.levels = ( [1e20] , )

# Change the colormap from rainbow to green to magenta.
v.setcolormap('grn_to_magenta')
print "Press the Return key to see next plot."
sys.stdin.readline()

# Change the colormap back to rainbow.
v.setcolormap('rainbow')

# Specify levels.
cf_asd.levels = ( [0,20],[20,40],[50,60],[60,70],[70,80],[90,100])

cf_asd.fillareastyle='pattern' # Change fill style to pattern
print "Press the Return key to see next plot."
sys.stdin.readline()
cf_asd.fillareastyle='hatch' # Change fill style to hatch
print "Press the Return key to see next plot."
sys.stdin.readline()
cf_asd.fillareaindices=([1,3,5,6,9,20]) # Hatch pattern
cf_asd.fillareacolors=([22, 44, 88, 122, 144, 188]) # Change color
print "Press the Return key to see next plot."
```

```
sys.stdin.readline()
# Create a persistent isofill graphics methods from an existing isofill graphics method.

cf_new = v.createisofill( 'new', 'ASD' ) # create new from ASD
cf_new2 = v.createisofill( 'new2','quick' )# create new2 from quick
cf_new.list()                                # list its attributes
v.show('isofill')                            # show isofill list with new and new2
v.removeobject( cf_new )                     # remove new from isofill list
v.show('isofill')                            # show isofill list without new
v.removeobject( cf_new2 )                     # remove new2
v.show('isofill')                            # show isofill list
```